



MS28076.02/MSFTP279USA

CERTIFICATE OF MAILING

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: **Mail Stop Appeal Briefs – Patents**, Assistant Commissioner for Patents, U.S. Patent and Trademark Office, Washington, D.C. 20231.

Date: 10-4-04


Himanshu S. Amin

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Applicant(s): David J. Kurlander, *et al.*

Serial No: 09/437,560

Filing Date: November 10, 1999

Examiner: Namitha Pillai

Art Unit: 2173

Title: METHOD AND SYSTEM FOR GENERATING USER-INTERFACE OUTPUT SEQUENCES

**Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450**

APPEAL BRIEF

Dear Sir:

Applicants' representative submits this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP279USA].

I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))

Appellants, appellants' legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))

Claims 1-13 and 23-25 have been cancelled. Claims 14-22 and 26-48 stand rejected by the Examiner. The rejection of claims 14-22 and 26-48 is being appealed.

IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))

No claim amendments have been entered after the Final Office Action.

V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))**A. Independent Claim 14**

Independent claim 14 relates to a method to facilitate a data processing system that includes a display device and a processing means for running an application program, wherein the application program has a user interface with a plurality of User Interface Output States and a plurality of operators, each operator utilized to transform a currently displayed User Interface Output State to a subsequent User Interface Output State. The method as recited in the subject independent claim comprises providing a user interface output system for controlling the generation of a user interface output sequence, providing a specification identifying a plurality of goal User Interface Output States for the user interface output system to establish and identify a plurality of operators, each of the operators having at least one precondition to be satisfied prior to the operation being performed, compiling the specification to provide a user interface output controller

distinct from the application program, wherein the resultant user interface output controller includes a plurality of plans, each plan having a series of operators, a start User Interface Output State and a goal User Interface Output State. The series of operators transforms the start User Interface Output State into at least one intermediate User Interface Output State and subsequently to the goal User Interface Output State, such that the preconditions of each operator in the series is satisfied after the performance of the earlier operators in the series. In addition, the method provided by independent claim 14 recites that while the application program is running on the processing means, the user interface output controller is provided an event identifying one of the goal User Interface Output States from the application program, determines the currently displayed User Interface Output State, retrieves one of the plurality of plans such that the start User Interface Output State of the retrieved plan is the currently displayed User Interface Output State and the goal User Interface Output State of the retrieved plan is the goal User Interface Output State identified by the event, and performs the series of operators provided by the retrieved plan to display the start User Interface Output State followed by at least one intermediate User Interface Output State followed by the goal User Interface Output State. (*See e.g.*, page 3, line 14-page 4, line 9).

B. Independent Claim 22

Independent claim 22 recites a method to be practiced in a data processing system including a display device and a processing means running an application program comprising providing a compiled user interface output controller distinct from the application program for generating a user interface output sequence, wherein the user interface output controller includes a first User Interface Output State and a second User Interface Output State; the first and second User Interface Output States each including a set of conditions representing values which capture attributes of the respective User Interface Output State. Further, the subject independent claim provides that under the control of the user interface output controller, the user interface output controller receives operators from the application program wherein each operator has a precondition consisting of one of the conditions in the set and a required value for the condition such that the operator can only be performed when a current User Interface Output State

satisfies the precondition by including the condition representing the required value. Upon receipt of the operators, the user interface output controller receives an event from the application program specifying a goal to be achieved by the user interface output sequence, whereupon the user interface output controller determines the conditions that temporally precede the event and then establishes the conditions so determined. The user interface output controller then performs the plurality of received operators thereby transforming the first User Interface Output State into the second User Interface Output State which establishes the event, wherein the first operator of the plurality of operators has a precondition which is satisfied by the current User Interface Output State and after performance of each operator that comprises the plurality of operators the resulting User Interface Output State satisfies the precondition for the operator subsequent in the plurality of operators. In addition, the user interface output controller determines whether conditions exist that temporally follow the event, and if so, the user interface output controller establishes the determined conditions that follow the event. (*See e.g.*, page 4, lines 10-25).

C. Independent Claim 26

Independent claim 26 recites a data processing system that comprises a display device for displaying a sequence of a plurality of User Interface Output States and a processing means for running an application program. (*See e.g.*, page 3, lines 15-16; page 4, lines 26-30; page 5, lines 8-11). The system also includes means for providing a user interface output system for controlling the generation of the sequence. (*See e.g.*, page 3, lines 16-17; page 5, lines 11-13; page 6, lines 18-20). Further, the system includes means for providing a specification identifying goal User Interface Output States for the user interface output system to establish and identifying a plurality of operators, each operator for transforming one User Interface Output State into another User Interface Output State such that a precondition of the operator is established by the one User Interface Output State and such that a post-condition of the operator is established in another User Interface Output State. (*See e.g.*, page 3, line 27-page 4, line 9; page 5, lines 13-16; page 7, lines 4-7; page 8, lines 22-33). Additionally, the system provides means for compiling the specification to generate a user interface output controller

distinct from the application program. (*See e.g.*, page 3, lines 30-32; page 5, lines 16-18; page 7, lines 23-27; page 15, line 24-page 16, line 2). In addition, the system provides means for storing the user interface output controller in memory, the user interface output controller including, means for receiving an event from the application program, the event identifying one of the goal User Interface Output States. (*See e.g.*, page 5, lines 18-21; page 7, lines 16-20). Moreover, the system provides means for determining a current User Interface Output State in the sequence. (*See e.g.*, page 3, lines 32-35; page 4, lines 3-4; page 5, lines 21-23; page 15, lines 31-35; page 20, lines 14-16). The system also provides means for determining a series of operators which transform the determined current User Interface Output State into the identified one of the goal User Interface Output State. (*See e.g.*, page 3, lines 32-36; page 4, lines 6-9; page 5, lines 23-25; page 15, line 35-page 16, line 2; page 20, lines 16-20). The system further includes means for performing the series of operators to display the sequence on the display device, the performing to transform the determined current User Interface Output State into at least one intermediate User Interface Output State and then into the identified one of the goal User Interface Output States. (*See e.g.*, page 3, lines 22-24; page 4, lines 6-9; page 5, lines 3-7; page 5, lines 25-29; page 7, lines 29-31; page 15, line 35-page 16, line 2; page 20, lines 7-21).

The means for limitations described above are identified as limitations subject to the provisions of 36 U.S.C. §112 ¶6. The structures corresponding to these limitations are identified with reference to the specification and drawings in the above-noted parentheticals.

D. Independent Claim 29

Independent claim 29 relates to a user interface output system for controlling the generation of a user interface output sequence. The system includes a specification for identifying goal user interface output states, which identify user interface output states for the user interface output system to establish and for identifying operators which specify actions to be performed by the user interface output sequence each of the operators having at least one precondition to be satisfied before the operator can be executed. (*See e.g.*, page 3, line27-page 4, line 9; page 5, lines 13-16; page 7, lines 4-5). The system

further includes a compiler for compiling the specification to generate a user interface output controller distinct from an application program. (*See e.g.*, page 3, lines 30-32; page 5, lines 16-18; page 7, lines 23-27; page 15, line 24-page 16, line 2). In addition, the system includes a storage for storing the user interface output controller in memory. (*See e.g.*, page 5, lines 18-19; page 7, lines 16-17). Further, the user interface output controller comprises, a receiver for receiving an event from the application program, the event identifying one of the goal user interface output states; (*See e.g.*, page 3, lines 16-20; page 3, line 36-page 4, line 3; page 4, lines 16-19; page 4, lines 28-36; page 5, lines 19-21; page 6, lines 20-24; page 6, line 33-page 7, line 3; page 7, lines 25-28). The user interface output controller also comprises a first determinor for determining a current user interface output state in the user interface output sequence. (*See e.g.*, page 3, lines 32-35; page 4, lines 3-4; page 5, lines 21-23; page 15, lines 31-35; page 20, lines 14-16). In addition, the user interface output controller also comprises a second determinor for determining a sequence of operators which transform the determined current user interface output state into at least one intermediate user interface output state and then into the identified one of the goal user interface output states, the operators in the sequence such that after execution of each of the operators in the sequence other than a last operator, the preconditions of a next operator in the sequence are satisfied. (*See e.g.*, page 3, lines 32-36; page 4, lines 6-9; page 5, lines 23-25; page 15, line 35-page 16, line 2; page 20, lines 16-20). Moreover, the user interface output controller includes an executor for executing the sequence of operators to transform the determined current user interface output state into the at least one intermediate user interface output state and then into the identified one of the goal user interface output states so as to display the sequence of operators on a display device. (*See e.g.*, page 3, lines 22-24; page 4, lines 6-9; page 5, lines 3-7; page 5, lines 25-29; page 7, lines 29-31; page 15, line 35-page 16, line 2; page 20, lines 7-21).

The limitations described above are identified as limitations subject to the provisions of 36 U.S.C. §112 ¶6. The corresponding structures are identified with reference to the specification and drawings in the parentheticals above.

E. **Independent Claim 32**

Independent claim 32 recites a computer-readable storage medium, upon which is stored a compiled user interface output controller for generating a user interface output sequence, the user interface output controller distinct from an application program and performing the steps of: receiving an event from the application program, the event specifying a goal to be achieved by the user interface output sequence by displaying a series of a plurality of user interface output states; and upon receiving the event from the application program, generating the user interface output sequence for achieving the goal user interface output sequence including: a plurality of operators that if executed when a predefined set of conditions are true will display the plurality of user interface output states, the predefined set of conditions including at least one precondition of a first of the plurality of operators such that at least one precondition must be true before the first operator can be executed, the operators in the sequence ordered such that execution of previous operators in the sequence will establish as true conditions necessary for execution of a next operator in the sequence; and executing the plurality of operators of the generated user interface output sequence when the predefined set of conditions is true so as to display the series of the plurality of user interface output states on a display device. (*See e.g.*, page 5, lines 8-29).

F. Independent Claim 33

Independent claim 33 recites a method practiced in a data processing system including a display device and a processing means running an application program, the application program having a user interface with a current User Interface Output State displayed on the display device, comprising: providing a compiled user interface output controller for displaying the user interface, the user interface output controller distinct from the application program; under control of the application program, without knowledge of the current User Interface Output State displayed on the display device, determining a goal User Interface Output State to be displayed on the display device; and sending to the user interface output controller an indication of the goal User Interface Output State; and under control of the user interface output controller, receiving from the application program the indication of the goal User Interface Output State; determining the current User Interface Output State; determining a sequence of a plurality of operators

based on the determined current User Interface Output State and the goal User Interface Output State; and for each of the operators in sequence, performing the operator to transition the user interface from the current User Interface Output State to a different resulting User Interface Output State, the performing such that the resulting User Interface Output State is displayed on the display device and becomes the current User Interface Output State and such that the current User Interface Output State after all the operators are performed is the goal User Interface Output State, the sequence such that the preconditions for a first operator in the sequence are currently established and such that the preconditions for all other operators in the sequence are established after execution of earlier operators in the sequence, whereby the application program specifies the goal User Interface Output State without knowledge of the current User Interface Output State, and in response the user interface output controller determines a sequence of operators that when performed transition the user interface from the current User Interface Output State through at least one displayed intermediate User Interface Output State to the displayed goal User Interface Output State. (*See e.g.*, page 6, line 18-page 7, line 31).

G. Independent Claim 37

Independent claim 37 relates to a method for displaying user interface information for a plurality of application programs, each application program having a distinct user interface and an operator set consisting of a plurality of operators, each operator having at least one precondition which must be established before the operator can be performed, the method comprising: providing a compiled user interface output controller distinct from the application programs; and under control of the user interface output controller and for each of the application programs, receiving from the application program the operator set for the application program; after receiving the operator set, receiving from the application program a plurality of user interface goals, each user interface goal reflecting information to be displayed in the user interface for the application program; and for each user interface goal, achieving the user interface goal by determining the information currently displayed in the user interface for the application program; after receiving the user interface goal, determining a series of operators from the operator set

for the application program that will transition the user interface for the application program from displaying the determined information to displaying the information reflected by the user interface goal, the series such that the preconditions for a first operator in the series are currently established and such that the preconditions for all other operators in the series are established after execution of earlier operators in the series; and executing the determined series of operators in sequence such that the information reflected by the user interface goal is displayed when the executing is complete, whereby an executing user interface output controller can receive a set of operators and a plurality of user interface goals from each of the plurality of application programs, and can determine for each user interface goal the series of operators from the operator set for the application program that when executed in sequence will achieve the user interface goal. (*See e.g.*, page 4, line 10-page 5, line 7).

H. Independent Claim 38

Independent claim 38 relates to a system that generates a user interface output controller, comprising: a compiler that compiles one or more specifications regarding a plurality of goal User Interface Output States; and a compiled user interface output controller, generated by the compiler, that is distinct from an application program. (*See e.g.*, page 3, line 30-32; page 7, lines 23-27; and page 15, line 24-page 16, line 2).

I. Independent Claim 43

Independent claim 43 recites a method for generating a user interface output controller comprising: retrieving at least one specification identifying at least one goal user interface output state for the user interface output controller to establish, the at least one specification further comprising at least one operator for specifying actions to be performed by the compiled user interface output controller, each operator having at least one precondition to be satisfied before the operator can be executed; and compiling the at least one specification to create a user interface output controller distinct from an application program. (*See e.g.*, page 3, line 25-page 4, line 25).

J. Independent Claim 48

Independent claim 48 relates to a system that generates a user interface output controller. The system includes means for compiling one or more specifications regarding a plurality of goal User Interface Output States. (*See e.g.*, page 3, lines 30-32; page 5, lines 16-18). Further, the system also includes means for generating a compiled user interface output controller that is distinct from an application program. (*See e.g.*, page 7, lines 23-27; page 15, line 24-page 16, line 2).

The means for limitations described above are identified as limitations subject to the provisions of 36 U.S.C. §112 ¶6. The corresponding structures are identified with reference to the specification and drawings in the parentheticals above corresponding to those claim limitations.

VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))

A. Claims 14-22 and 26-48 are unpatentable under 35 U.S.C. §102(e) over Peterson *et al.* (US 5,801,687).

VII. Argument (37 C.F.R. §41.37(c)(1)(vii))

A. Rejection of Claims 14-22 and 26-48 Under 35 U.S.C. §102(e)

Claims 14-22 and 26-48 stand rejected under 35 U.S.C. §102(e) as being anticipated by Peterson *et al.* (US 5,801,687). Withdrawal of this rejection is respectfully requested for at least the following reasons.

i. Peterson et al. fails to disclose, teach or suggest each and every limitation set forth in the subject claims.

A single prior art reference anticipates a patent claim only if it expressly or inherently describes *each and every limitation set forth in the patent claim*. *Trintec Industries, Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The *identical invention must be shown in as complete detail as is contained in the ...*

claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added).

Independent claims 14, 22, 26, 29, 32, 33, 37, 38, 43 and 48 recite similar claim limitations: *a compiler for compiling the specification which results in a user interface output controller distinct from the application program*. Thus, it is apparent that the invention as claimed utilizes a *compiler* to compile a specification into *a user interface output controller distinct from an application program*. Peterson *et al.* fails to disclose this particular novel feature of the claimed invention.

Peterson *et al.* discloses an authoring tool comprising at least one nestable graphic state and a state machine, wherein each state machine comprises one or more states and zero or more transitions, each transition interconnecting a first state with a second state. See Abstract. The invention disclosed by Peterson *et al.* in essence, provides a software tool for *creating and/or modifying multimedia products on a computer system*, thus allowing a user of the software tool to *view a state machine simultaneously in several different formats*, and further allows the user, through an interface, *to manipulate a map of permissible states to add, delete or modify states and transitions*. See, col. 5, line 38-col. 6, line 11. This is in stark contrast to applicants' invention as claimed, which provides a user interface output system for controlling the generation of user interface output sequences. The user interface output system as disclosed by applicants provides event definitions, which specify, at a high-level, a goal to be achieved by the user interface output sequence communicated from an application program. When the user interface output system receives an event from the application program, the user interface output system issues low-level commands to direct the actions of the user interface output sequence. The application program can then ignore the details of invoking the low-level commands and may request an event in order to have the user interface output sequence achieve a goal. Thus, the claimed invention simplifies the process of generating complex user interface output, as well as combining various forms of user interface output, such as animation, audio, and textual elements.

In order to effectuate the invention as claimed, a specification is provided which identifies goal user interface output states, wherein the goal user interface output states

identify user interface output states for the user interface output system to establish. The specification also identifies operators that specify actions to be performed by the user interface output sequence. In addition, a compiler is provided to compile the specification, which results in a user interface output controller distinct from the application program,. The resultant user interface output controller includes plans, wherein each plan has a series of operators to transform a start user interface output state into a goal user interface output state, a goal user interface output state, and a start user interface output state that identifies a current user interface output state of the user interface output sequence. Thus, while an application program is executing, the executing application program need only supply to the resultant compiled user interface output controller an event identifying one of the goal user interface output states, whereupon the compiled user interface output controller determines the appropriate start state of the user interface output sequence, retrieves the plan associated with the determined start state of the user interface output sequence and one of the identified the goal user interface output states, and performs the series of operators provided by the retrieved plan to transform the determined start user interface output state into the identified goal user interface output state so as to display the series of operators on a display device.

In the Final Office Action dated April 6, 2004, the Examiner contended that Peterson *et al.* provides a compiler for compiling the specification with the resultant distinct user interface output controller at col. 6, lines 1-5, which states:

For example, an authoring tool can allow an author to select an arbitrary state machine from within a multimedia product and, in response to such selection, can provide a representative map of the various states within the selected machine.

As was stated in applicants' Reply to the Final Office Action dated April 6, 2004, and is reiterated herein, Peterson *et al.*, rather than providing a compiler to compile a specification that results in a compiled user interface output controller distinct from the application, discloses a selection facility that allows an author to select arbitrary state machines from within a multimedia product, the authoring tool in response to the

selection providing a representative map of the various states within the selected state machine. It would thus appear that in Peterson *et al.* the nested state machines do not have an existence distinct from the multimedia product from which the state machines are selected, and that the representative map of the various states within the selected machine is a function incorporated into the authoring tool. Thus, applicants' representative avers that Peterson *et al.* fails to disclose, teach or suggest *a compiler for compiling the specification which results in a user interface output controller distinct from the application program.*

Moreover, the Examiner in the Advisory Action dated July 29, 2004, stated: "The disclosure of a compiler used in compiling a set of data that provides a set of instructions for determining the output, wherein these set of instructions as disclosed in the claims are very *similar* to a state machine which would provide a means for a controller." *Id.* (emphasis added). The Examiner is reminded that the standard by which anticipation is judged is not *similarity*, but rather *strict identity*, i.e., "the *identical invention must be shown in as complete detail as is contained in the patent claim.*" Richardson at 1236 (emphasis added). Thus, the fact that Peterson *et al.* might disclose a *similar* facility to that which is disclosed by the applicants, does not meet the sufficiency necessary to comport with the requirement under 35 U.S.C. §102, that a *strict identity* be established between the cited document and the invention as claimed.

In addition, in the Advisory Action dated July 29, 2004, the Examiner intimates that the "reference clearly teaches the use of state machines, wherein a set of instructions are used that rely on inputs and current outputs to follow a sequence to determine the output that will occur to the screen based on following these instructions, wherein this would represent the 'user interface output controller'." While applicants' representative agrees with the Examiner's characterization of Peterson *et al.* as disclosing the use of state machines, applicants' representative disagrees with the Examiner's assertion that "a set of instructions are used that rely on inputs and current outputs to follow a sequence to determine the output that will occur to the screen based on following these instructions, wherein this would represent the 'user interface output controller'," for the following reason. While it is widely recognized that a state machine is an abstraction that consists of a set of states that includes an initial state, a set of input events, a set of output events

and transition operators that transform the current state and input event into a new set of output events and the next state. The Examiner's characterization, would by necessary implication encompass every and all compiled computer programs, as each compiled programming instruction is input that changes one or more states and may cause other actions to take place. It is thus submitted that while every compiled computer program can be reduced in the abstract to being envisioned as one or more interrelated state machines, and in many instances while a computer program is being developed can be conceptually visualized as a set of one or more state machines, the abstract concept depicted in the state machine, once programmed into a set of program instructions and compiled into machine code, ceases to be the abstraction embodied in the initial state machine of its conception; the act of converting a state machine through the use of a programming language and a compiler converts the state machine from its abstract form into a tangible entity – the computer program. Thus, a programming language and compiler is a necessity to transform an abstract state machine into compiled machine instructions, but the converse is never true; a state machine being merely an abstraction or visual representation of permissible states and transitions does not require a compiler to transform it into something else; a state machine always remains merely an abstraction.

The Examiner further contended in the Final Office Action dated April 6, 2004, that:

As seen in Figure 3 [of Peterson *et al.*], there is a distinction shown between this software title application and the “TOOL” component which serves as the compiler that is responsible for compiling the necessary data for displaying the output . . . *Id.* at page 16.

While applicants' representative agrees that a distinction is shown in Figure 3 between a software title and the “TOOL”, applicants' representative nevertheless asserts that nowhere in Peterson *et al.* is it expressly or inherently disclosed that the “TOOL” comprises *a compiler* that compiles the specification which results *in a user interface output controller distinct from the application program*. Further, according to the commentary associated with Figure 3, Figure 3 illustrates the layers of *software* incorporated into the computer system when the invention is incorporated into an

authoring tool for multimedia products, wherein the *software* comprises: an operating system for controlling and coordinating the computer system, the operating system, in conjunction with the computer on which it is running, includes the capability to process sound, graphics, video or animation and to provide a windowing environment for display on the display screen; a software development environment that conceptually lies between the operating system and the tool, thereby providing an interface between the operating system and the tool; a tool, the subject of Peterson *et al.*'s invention; and one or more multimedia titles. Thus, even though there may be a distinction between the software title and the "TOOL", Peterson *et al.* is silent regarding the provision of a mechanism to compile the specification into the resultant user interface output controller distinct from the application program.

In the Advisory Action dated July 29, 2004, the Examiner asserted: "a compiler would be an inherent process, and as disclosed in the reference wherein a step would be needed for transforming any set of data into a set of instructions or to 'compile' together a set of data to be used as the 'user interface output controller'." *Id.* Applicants' representative avers to the contrary, as the so-called inherency is not warranted in Peterson *et al.* for the following reasons. It appears that Peterson *et al.* is confined to the viewing of state machines and the manipulation and creation of multimedia products. Thus, the totality of state machines subject to manipulation by the authoring tool disclosed in Peterson *et al.* are already existent in the multimedia product prior to the multimedia product being manipulated or created by Peterson *et al.*'s authoring tool. Moreover, if *assuming arguendo*, there were a compilation process inherent in Peterson *et al.* the result would not be a *user interface output controller*, but rather another multimedia product. Thus, Peterson *et al.* is clearly distinguishable from the invention as claimed in that no compilation facility is inherent therein, and if, *arguendo*, there were an inherent compilation facility, the result of the compilation would not be the *user interface output controller* as recited in the subject claims, but rather *a multimedia product*.

Further in the Advisory Action dated July 29, 2004, the Examiner stated: "The concept of using a compiler to develop a set of data to be used is quiet [sic] common in the computing field as disclosed in the reference." *Id.* As applicants' representative

acknowledged in the Reply to the Advisory Action, the utilization of compilers is well known in the art, but as applicants' representative stated in the Reply, and reiterates herein, compilers are always utilized to translate source code into object code, i.e., human-readable code into machine code. Compilers are never used to "develop a set of data" as the Examiner seems to claim. Moreover, while the Examiner claims that the "concept of using a compiler ... is disclosed in the reference", the Examiner has been unable to provide an indication where specifically this disclosure might be found within the cited document, which leads one to the inexorable conclusion that Peterson *et al.* fails to disclose the utilization of ***a complier to compile the specification which results in a user interface output controller distinct from the application program.***

Accordingly, in view of at least the forgoing, it is respectfully submitted that Peterson *et al.* fails to disclose, teach or suggest each and every limitation recited in independent claims 14, 22, 26, 29, 32, 33, 37, 38, 43 and 48, and claims that depend therefrom, and consequently that this reject should be withdrawn.

B. Conclusion

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 14-22 and 26-48 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,
AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24th Floor, National City Center
1900 East 9th Street
Telephone: (216) 696-8730
Facsimile: (216) 696-8731

VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))

1–13. Cancelled

14. In a data processing system including a display device and a processing means running an application program, the application program having a user interface with a plurality of User Interface Output States (UIOSes) and a plurality of operators, each operator for transforming a currently displayed UIOS to a displayed next UIOS, a method comprising the steps of:

providing a user interface output system for controlling the generation of a user interface output sequence;

providing a specification identifying a plurality of goal UIOSes for the user interface output system to establish and identifying the plurality of operators, each of the operators having at least one precondition to be satisfied before the operator can be performed;

providing a compiler for compiling the specification which results in a user interface output controller distinct from the application program, the user interface output controller including a plurality of plans, each of the plans having a series of operators, a start UIOS and one of the goal UIOSes, the series of operators for transforming the start UIOS to at least one intermediate UIOS to the goal UIOS, the operators in the series such that the preconditions of each of the other operators in the series are satisfied after performance of earlier operators in the series; and

while the application program is running on the processing means,

providing the user interface output controller with an event received from the application program, the event identifying one of the goal UIOSes;

determining a currently displayed UIOS;

retrieving one of the plurality of plans such that the start UIOS of the retrieved plan is the currently displayed UIOS and the goal UIOS of the retrieved plan is the goal UIOS identified by the event; and

performing the series of operators provided by the retrieved plan to display the start UIOS followed by the at least one intermediate UIOS followed by the goal

UIOS.

15. The method of claim 14 wherein the step of providing a specification includes the steps of:

providing events which may be specified by the application program and for which the user interface output system provides user interface output sequences;

providing state variables that define attributes of each user interface output state; and

providing operators that identify actions which are used to modify the attributes of each user interface output state.

16. The method of claim 15 wherein the specification includes timing directives which determine the time at which an action is performed.

17. The method of claim 15 further comprising providing state class definitions which are hierarchical groupings of state variables.

18. The method of claim 15 further comprising providing autonomous action sequences identifying actions which are performed by the user interface output system when a current user interface output state contains predefined values for conditions which capture attributes of the current user interface output state.

19. The method of claim 14 wherein the step of compiling the specification applies a planning methodology to generate each of the plans.

20. The method of claim 19 wherein the step of applying a planning methodology to generate each of the plans includes the steps of

selecting each of the plurality of goal UIOSes;

for each selected goal UIOS, selecting each of the plurality of operators:

performing an inverse of the selected operator on the selected goal UIOS; and

when the operators transforms the selected goal UIOS into a new UIOS, storing the new UIOS along with the selected operator.

21. The method of claim 20 wherein the new UIOS is identified as an intermediate UIOS and is then processed as a goal UIOS.

22. In a data processing system including a display device and a processing means running an application program, a method comprising the steps of:

providing a compiled user interface output controller for generating a user interface output sequence, the user interface output controller distinct from the application program, the user interface output sequence including a first user interface output state (UIOS) and a second user interface output state, the first user interface output state and the second user interface output state each including a set of conditions representing values which capture attributes of that user interface output state; and

under the control of the user interface output controller, receiving operators from the application program each operator having a precondition consisting of one of the conditions in the set and a required value for the condition such that the operator can only be performed when a current user interface output state satisfies the precondition by including the condition representing the required value;

after receiving the operators, receiving an event from the application program specifying a goal to be achieved by the user interface output sequence;

upon receiving the event from the application program, determining conditions which temporally precede the event;

establishing the determined conditions which precede the event;

performing a plurality of the received operators to transform the first user interface output state into the second user interface output state, which establishes the event, the plurality such that a first operator of the plurality has a precondition which is satisfied by a current user interface output state and wherein after the performance of each operator in the plurality resulting UIOS satisfies the precondition for the operator next in the plurality;

determining conditions which temporally follow the event, and establishing the determined conditions which follow the event.

23–25. Cancelled

26. A data processing system, comprising:
 - a display device for displaying a sequence of a plurality of user interface output states (UIOSes);
 - a processing means for running an application program;
 - means for providing a user interface output system for controlling the generation of the sequence;
 - means for providing a specification identifying goal UIOSes for the user interface output system to establish and identifying a plurality of operators, each operator for transforming one UIOS into another UIOS such that a precondition of the operator is established by the one UIOS and such that a postcondition of the operator is established in another UIOS;
 - means for compiling the specification to generate a user interface output controller distinct from the application program; and
 - means for storing the user interface output controller in memory, the user interface output controller including, means for receiving an event from the application program, the event identifying one of the goal UIOSes
 - means for determining a current UIOS in the sequence;
 - means for determining a series of operators which transform the determined current UIOS into the identified one of the goal UIOS and
 - means for performing the series of operators to display the sequence on the display device, the performing to transform the determined current UIOS into at least one intermediate UIOS and then into the identified one of the goal UIOSes.
27. The system of claim 26 wherein the received event identifies a timing specification which determines the time at which the series of operators are performed.

28. The system of claim 27 further comprising means for incorporating the identified timing specification into the sequence.

29. A user interface output system for controlling the generation of a user interface output sequence, comprising:

a specification for identifying goal user interface output states, which identify user interface output states for the user interface output system to establish and for identifying operators which specify actions to be performed by the user interface output sequence each of the operators having at least one precondition to be satisfied before the operator can be executed;

a compiler for compiling the specification to generate a user interface output controller distinct from an application program; and

a storage for storing the user interface output controller in memory, the user interface output controller comprising,

a receiver for receiving an event from the application program, the event identifying one of the goal user interface output states;

a first determinor for determining a current user interface output state in the user interface output sequence;

a second determinor for determining a sequence of operators which transform the determined current user interface output state into at least one intermediate user interface output state and then into the identified one of the goal user interface output states, the operators in the sequence such that after execution of each of the operators in the sequence other than a last operator, the preconditions of a next operator in the sequence are satisfied; and

an executor for executing the sequence of operators to transform the determined current user interface output state into the at least one intermediate user interface output state and then into the identified one of the goal user interface output states so as to display the sequence of operators on a display device.

30. The system of claim 29 wherein the received event identifies a timing specification which determines the time at which the sequence of operators are performed.
31. The system of claim 30 further comprising a timer for determining the time at which the sequence of operators are performed.
32. A computer-readable storage medium, upon which is stored a compiled user interface output controller for generating a user interface output sequence, the user interface output controller distinct from an application program and performing the steps of:
 - receiving an event from the application program, the event specifying a goal to be achieved by the user interface output sequence by displaying a series of a plurality of user interface output states; and
 - upon receiving the event from the application program, generating the user interface output sequence for achieving the goal user interface output sequence including: a plurality of operators that if executed when a predefined set of conditions are true will display the plurality of user interface output states, the predefined set of conditions including at least one precondition of a first of the plurality of operators such that at least one precondition must be true before the first operator can be executed, the operators in the sequence ordered such that execution of previous operators in the sequence will establish as true conditions necessary for execution of a next operator in the sequence; and
 - executing the plurality of operators of the generated user interface output sequence when the predefined set of conditions is true so as to display the series of the plurality of user interface output states on a display device.
33. In a data processing system including a display device and a processing means running an application program, the application program having a user interface with a current User Interface Output State (UIOS) displayed on the display device, a method comprising:

providing a compiled user interface output controller for displaying the user interface, the user interface output controller distinct from the application program; under control of the application program, without knowledge of the current UIOS displayed on the display device, determining a goal UIOS to be displayed on the display device; and sending to the user interface output controller an indication of the goal UIOS; and under control of the user interface output controller, receiving from the application program the indication of the goal UIOS; determining the current UIOS; determining a sequence of a plurality of operators based on the determined current UIOS and the goal UIOS; and

for each of the operators in sequence, performing the operator to transition the user interface from the current UIOS to a different resulting UIOS, the performing such that the resulting UIOS is displayed on the display device and becomes the current UIOS and such that the current UIOS after all the operators are performed is the goal UIOS, the sequence such that the preconditions for a first operator in the sequence are currently established and such that the preconditions for all other operators in the sequence are established after execution of earlier operators in the sequence,

whereby the application program specifies the goal UIOS without knowledge of the current UIOS, and in response the user interface output controller determines a sequence of operators that when performed transition the user interface from the current UIOS through at least one displayed intermediate UIOS to the displayed goal UIOS.

34. The method of claim 33 wherein the determined sequence of the plurality of operators is generated after the receiving of the indication of the goal UIOS.

35. The method of claim 33 wherein each UIOS has a value for each of a plurality of UIOS variables, wherein each operator has a precondition of a UIOS variable and a required value for the UIOS variable, wherein an operator having a precondition of a first variable and a first value can only be performed when the current UIOS satisfies the operator precondition by having a value for the first variable that is equal to the first

value, wherein the determined sequence begins with a first operator whose precondition is satisfied by the determined current UIOS, and wherein after the performance of each operator in the determined sequence, the resulting UIOS satisfies the precondition for the operator next in the determined sequence.

36. The method of claim 35 wherein the determined sequence is generated by:
 - identifying the UIOS variable whose value in the goal UIOS is different than in the current UIOS;
 - selecting a last operator such that the resulting UIOS from performing the last operator has a value for the identified UIOS variable that is equal to the value for the identified UIOS variable in the goal UIOS;
 - selecting an initial operator such that the resulting UIOS from performing the initial operator satisfies the precondition of the last operator;
 - until the current UIOS satisfies the precondition of the initial operator, repeatedly performing the steps of
 - designating the initial operator to be an intermediary operator; and
 - selecting an initial operator such that the resulting UIOS from performing the initial operator satisfies the precondition of the operator most recently designated to be the intermediary operator;
 - determining the plurality of operators to be the selected operators; and
 - determining the sequence of the plurality of operators to be a reverse of the selection order.

37. A method for displaying user interface information for a plurality of application programs, each application program having a distinct user interface and an operator set consisting of a plurality of operators, each operator having at least one precondition which must be established before the operator can be performed, the method comprising:
 - providing a compiled user interface output controller distinct from the application programs; and
 - under control of the user interface output controller and for each of the application programs,

receiving from the application program the operator set for the application program;

after receiving the operator set, receiving from the application program a plurality of user interface goals, each user interface goal reflecting information to be displayed in the user interface for the application program; and

for each user interface goal, achieving the user interface goal by determining the information currently displayed in the user interface for the application program;

after receiving the user interface goal, determining a series of operators from the operator set for the application program that will transition the user interface for the application program from displaying the determined information to displaying the information reflected by the user interface goal, the series such that the preconditions for a fast operator in the series are currently established and such that the preconditions for all other operators in the series are established after execution of earlier operators in the series; and

executing the determined series of operators in sequence such that the information reflected by the user interface goal is displayed when the executing is complete,

whereby an executing user interface output controller can receive a set of operators and a plurality of user interface goals from each of the plurality of application programs, and can determine for each user interface goal the series of operators from the operator set for the application program that when executed in sequence will achieve the user interface goal.

38. A system that generates a user interface output controller, comprising:

a compiler that compiles one or more specifications regarding a plurality of goal UIOSes; and

a compiled user interface output controller, generated by the compiler, that is distinct from an application program.

39. The system claim of claim 38, the compiled user interface output controller comprises:

an input component to receive an event from the application program, the event identifies one or more goal user interface output states;

a first determinor that determines a current user interface output state in a user interface output sequence;

a second determinor that determines a sequence of operators which transform the determined current user interface output state into at least one intermediate user interface output state and then into the identified goal user interface output state;

a sequencer that comprises a sequence of operators that after execution of each operator in sequence other than the last operator, the precondition of a next operator in the sequence is satisfied; and

an executor that executes the sequence of operators to transform the determined current user interface output state into the at least one intermediate user interface output state and then into the identified goal user interface output state so as to display the sequence of operators on a display device.

40. The system of claim 39, the received event identifies a timing specification that determines the time in which the sequence of operators are performed.

41. The system of claim 40, that comprises a timer that determines the time in which the sequence of operators are performed.

42. The system claim of claim 38, that further comprises a storage that stores the compiled user interface output controller in memory.

43. A method for generating a user interface output controller comprising:
retrieving at least one specification identifying at least one goal user interface output state for the user interface output controller to establish, the at least one specification further comprising at least one operator for specifying actions to be performed by the compiled user interface output controller, each operator having least one precondition to be satisfied before the operator can be executed; and
compiling the at least one specification to create a user interface output controller

distinct from an application program.

45. The method of claim 43, further comprising identifying a timing specification which determines the time in which the sequence of operators is to be performed.

46. The method of claim 45 further comprising determining the time in which the sequence of operators are performed.

47. The method claim of claim 43, further comprising storing the compiled user interface output controller in memory.

48. A system that generates a user interface output controller, comprising:
means for compiling one or more specifications regarding a plurality of goal UIOSes; and
means for generating a compiled user interface output controller that is distinct from an application program.

IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))

None.

X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))

None.